DOCUMENT RESUME

ED 101 688 IR 001 538

AUTHOR Roman, Richard A.; Laudato, Nicholas C.
TITLE Computer Assisted Instruction in Word Problems: Rationale and Design.
INSTITUTION Pittsburgh Univ., Pa. Learning Research and Development Center.
SPONS AGENCY National Inst. of Education (DHEW), Washington, D.C.; National Science Foundation, Washington, D.C.
REPORT NO PU-LRDC-1974-19
PUB DATE 74
NOTE 46p.

EDRS PRICE MF-$0.76 HC-$1.95 PLUS POSTAGE
DESCRIPTORS Arithmetic Curriculum; *Computer Assisted Instruction; Educational Research; Elementary Education; Information Processing; Instructional Design; *Learning Difficulties; *Problem Solving
IDENTIFIERS Word Problems

ABSTRACT

A Word Problem Program was set up to teach elementary school students how to solve arithmetic word problems and to teach some problem solving skills applicable to nonword problems. All instruction is accomplished through a computer program. Goals of the program are described along with three basic instructional decisions and the reasons for them, and the student's interaction with the program. The instructional design is detailed, along with samples of student interaction and a comparison of pilot and complete versions of the program. (SK)

ED101688

COMPUTER ASSISTED INSTRUCTION IN WORD PROBLEMS: RATIONALE AND DESIGN

Richard A. Roman and Nicholas C. Laudato

Learning Research and Development Center

University of Pittsburgh

1974/19

U S DEPARTMENT OF HEALTH, EDUCATION & WELFARE
NATIONAL INSTITUTE OF EDUCATION
THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

IR 001 538

The research reported herein was supported by a grant from the National Science Foundation (NSF-GJ-540X) and by the Learning Research and Development Center, supported in part as a research and development center by funds from the National Institute of Education (NIE), United States Department of Health, Education, and Welfare. The opinions expressed in this publication do not necessarily reflect the position or policy of the sponsoring agencies and no official endorsement should be inferred.

2/3

TABLE OF CONTENTS

		Page
Section I:	The Problem	1
	A. Arithmetic Word Problems as a Problem Solving Task	2
	B. A Model of Word Problem Solving	4
Section II:	Instructional Design of the Word Problem Program	8
	A. The Role of Computation	9
	B. Instructional Strategies	10
	C. Defining and Sequencing the Instructional Objectives	13
	1. The Structural Variable Approach	14
	2. The Information Processing Approach	21
	3. A Combined Approach	23
Section III:	The Word Problem Program	31
	A. The Pilot Version, Spring 1974	33
	B. The Complete Version, Fall 1974	37
References		39

LIST OF FIGURES

Figure		Page
1.	Information Processing Model of Word Problem Solution	5
2.	A Simple Word Problem and Its Hints	12
3.	A Complex Word Problem and Its Hints	13
4.	Loftus Word Problem Format	16
5.	Loftus Response Format	17
6.	Example of an Item Form	25
7.	Steps in Translating an Item Form	26
8.	Array of Instructional Objectives for Problem Types	29
9.	One Student's Mastery Pattern	32
10.	An Initial Problem Display	33
11.	Display after One Computation	34
12.	An Incorrect Answer Display	35
13.	A Hint Display	35
14.	A Correct Answer Display	36

COMPUTER ASSISTED INSTRUCTION IN WORD PROBLEMS: RATIONALE AND DESIGN

Richard A. Roman and Nicholas C. Laudato

Learning Research and Development Center
University of Pittsburgh

Section I: The Problem

The Word Problem Program was designed to teach elementary school students how to solve arithmetic word problems. A second design objective was to teach some problem solving skills applicable in problem solving situations not involving word problems. All instruction in the Word Problem Program is accomplished through a computer program. This report presents the goals of the program, describes three instructional decisions and why they were made, and then describes the students' interaction with the computer program.

Section I presents the goals for the program, then describes a model of people solving word problems. The model includes two stages: (1) an initial translation of the word problem into symbolic notation, and (2) a solution stage that produces a numeric answer.

Section II presents the three major instructional decisions made about the program and provides a rationale for these decisions from psychological and educational research. The first decision was to focus attention on the translation stage of the model, assuming that students know how to perform the solution stage. The second decision was to combine elements of a formal analysis method with use of analogous problems to help students who have difficulty solving the problems as they are initially presented. The literature demonstrates that these

two methods teach effectively (Morton, 1925; Suydam & Riedesel, 1969; Gorman, 1967). The third decision was to create instructional objectives based on the number and kinds of information processing skills required in the translation stage. All problems in the same instructional objective require similar information processing steps. This approach contrasts with traditional sequencing by surface features of the problem.

Section III presents a sample interaction with a student, detailing how each of the instructional decisions of Section II was implemented in the Word Problem Program.

A. Arithmetic Word Problems as a Problem Solving Task

Many mathematicians regard arithmetic word problems as holding a crucial role in the curriculum. For example, G. Polya (1962) asserts that ". . . the most important single task of mathematics instruction in the secondary school is to teach the setting up of equations to solve word problems" (p. 59). He holds this view because the translation of a real situation into mathematical terms is the first opportunity offered by the curriculum for the basic experience "that mathematical concepts may be related to realities, but such relations must be carefully worked out." Since 1962, when Polya wrote, it has become common for elementary school curricula to include work with linear and quadratic number sentences with integer roots (Wooten, 1965; Educational Services, Inc., 1963). The basic translation experience Polya advocates can now be incorporated into elementary school mathematics. Word problems can create the crucial bridge between the world of abstract mathematical symbols and the world of concrete problem situations; however, the word problem instruction which most elementary school students receive rarely builds that bridge.

In most elementary school curricula, word problems are used as exercises to increase competence in computation. For example, a lesson designed to teach the student to multiply three-digit numbers will often be followed by a set of word problems which requires the multiplication of three-digit numbers; the student can solve the problem without reading a single word. In their report on research and development in elementary school mathematics, Suydam and Riedesel (1969) state that studies ". . . reveal that pupils often give little attention to the actual problems; instead they almost randomly manipulate numbers" (p. 47). R. P. Stevenson (cited in Riedesel, 1967) pointed out that manipulation of the numbers is more algorithmic than random. For example, he describes a method used by an elementary school student:

> If there are lots of numbers, I add. If there are only two numbers with lots of parts, I subtract. But if there are just two numbers with one littler than the other, it is hard. I divide if they come out even, but if they don't, I multiply (p. 308).

Many students, trained in conventional ways, respond to word problems on the basis of three features: (1) the number, magnitude, and divisibility of the numbers in the problem; (2) the operations used to solve the preceding problem; and (3) the individual words or phrases which denote operations. One goal of this work is to avoid reinforcing algorithms of the type described above, a point amplified in Section II.A below. A definition will clarify the meaning of "problem solving behavior."

This report adopts the information processing viewpoint, proposed by Newell and Simon (1972), as the definition of a person in a problem solving situation: He wants something, but does not immediately know the sequence of actions required to get it. For the situation to be a problem, the person must have a method available to test proposed

solutions to see if they solve the problem. A person solving a word problem wants to find the solution to a problem presented in words that requires a quantitative answer.

A person trying to solve any problem can make use of the general problem solving skills hypothesized by Polya (1957), Olton (1969), Newell and Simon (1972), and Papert (1972). These include: recognizing the existence of a problem solving situation and clearly stating a goal; gathering, recording, and organizing the data; breaking a complex problem into parts, solving the parts, and reintegrating the partial solutions into a solution to the whole problem; and recognizing when the solution has been attained, when help is needed, or when to give up.

B. A Model of Word Problem Solving

In addition to general problem solving skills, a person solving a word problem must exhibit a variety of skills specific to word problems. An expanded model of behavior in word problem solving situations, based upon the information processing work of Bobrow (1968) and Simon and Paige (1966), specifies these word problem skills. According to the model (Figure 1), people solve word problems in two stages: (1) translation stage in which people create a mathematical representation of the problem, and (2) solution stage in which people manipulate the mathematical representation to solve the problem. If a person fails to solve the problem in the solution stage, he can reinvoke the translation stage, adding more data to the mathematical representation, usually by adding assumptions about the problem.

In the model, each stage contains two interacting parts. The complete structure is represented in Figure 1. Each part will be described in terms of a sample word problem:

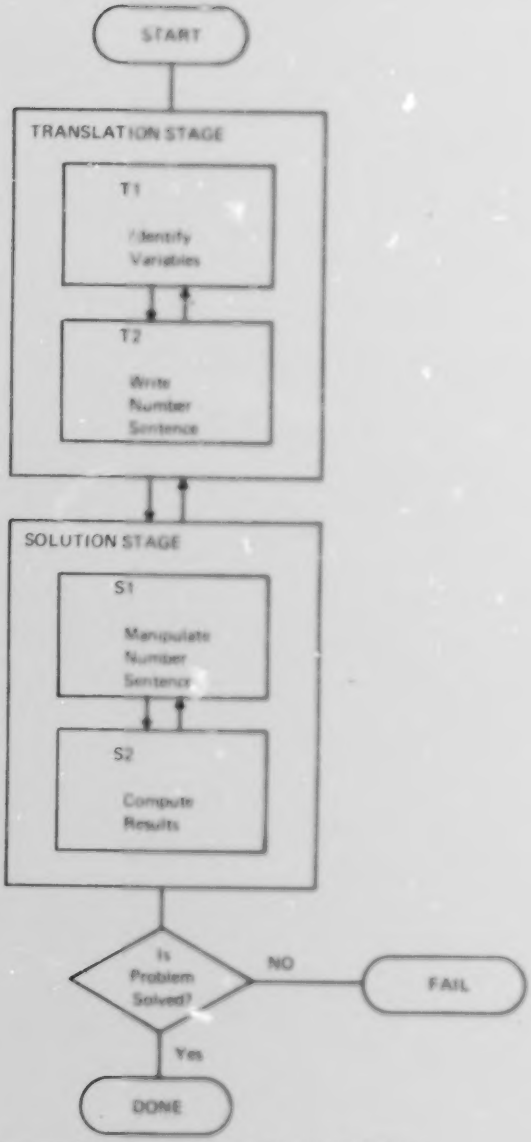

Figure 1. Information Processing Model of Word Problem Solution

ON THE WAY TO THE MARKET, JOHN FOUND 51 SEEDS.
JOHN FOUND 17 SEEDS MORE THAN MARY.
HOW MANY SEEDS DID MARY FIND?

The first part of the translation stage (T1 in Figure 1) associates the variables in the problem with symbols. In this problem, it creates three associations:

JOHN'S SEEDS = 51
JOHN'S SEEDS MORE THAN MARY'S = 17
MARY'S SEEDS = ?

The second part of the translation stage (T2 in Figure 1) combines the variables to produce a translation of the verbal word problem to number sentence form. This proceeds in several steps:

First, the simplified problem: JOHN'S SEEDS = JOHN'S SEEDS MORE THAN MARY'S + MARY'S SEEDS is created using general semantic knowledge about the meaning of "more than." The initial formation is transformed by direct substitution to: 51 = 17 + ?.

The first part of the solution stage (S1 in Figure 1) transforms the number sentence into a direct number sentence. In this case, 51 - 17 = ?. The second part of the solution stage (S2 in Figure 1) does the arithmetic computations required during or after the first part. In this case, the final result is ? = 34, indicating that 34 is the number of Mary's seeds.

Some people will transform the problem during the translation stage to create a direct number sentence. In this case, noting the semantic relationship between "more than" and "less than," they might create the simplified problem JOHN'S SEEDS - MARY'S SEEDS LESS THAN JOHN'S = MARY'S SEEDS instead of the one given above. These

semantic transformations do not change the model, and are described more fully in Section II.C.3, where the model is applied to a more complex problem.

While the language used to describe the model is algebraic, elementary school students, and many adults, do not use formal algebra to solve word problems. The algebraic language, then, is a metaphor for the processes of representation and manipulation which do actually take place.

The information processing model clarifies the skills required to solve word problems. The skills of the solution stage are algorithmic, while the skills of the translation stage include higher level cognitive processing. Since the Word Problem Program is concerned with problem solving and not with competence in algorithmic manipulation, only the skills of the translation stage are taught in the program. It is assumed that students using the Word Problem Program are capable of solving the number sentences upon which the word problems they encounter are based. Students are also assumed to be able to compute accurately. Some of the implications of this decision are discussed in Section II.A.

This section has examined the problem of teaching elementary school students to solve arithmetic word problems. It pointed out that many word problem curricula reinforce undesirable solution strategies that rely on such factors as the magnitude, number and divisibility of numbers in the problem, the operations used in the last problem, and single word cues within the problem. A distinction between such algorithmic behavior and problem solving behavior was drawn. Finally, an information processing model of people solving word problems was developed. The next section explores three instructional design decisions made while implementing the Word Problem Program.

Section II: Instructional Design of the Word Problem Program

Several features of the Word Problem Program are determined by the setting in which it is implemented. Before discussing the major design questions it is useful to describe that setting.

The Word Problem Program operates on the small computer resource at the Oakleaf Elementary School, a school which has cooperated in developing educational material with the Learning Research and Development Center since 1964. All mathematics work at Oakleaf is done on an individualized basis, and the Word Problem Program is, therefore, designed to teach one student at a time (Lindvall & Bolvin, 1966).

In Spring 1972, a small computer resource (a PDP 15 with 64K 18-bit words of memory) was installed in a van outside Oakleaf, and now provides locally controlled service to the school throughout the day. In spite of the computer's small size, it has a general-purpose time-sharing system (ETSS) designed and implemented by the Learning Research and Development Center (Fitzhugh, 1970). ETSS currently supports sixteen terminals.

Given the individualized instructional system and the computer resource, the Word Problem Program must teach individual students (rather than groups of students) instructional objectives at their level of competence. The overall instructional strategy is to present problems on the computer terminal, allowing the student to solve them and to ask for help when it is needed. The program judges the effectiveness of the solution and chooses new problems appropriate to the state of learning. Branching from objective to objective occurs when mastery (or nonmastery) of the current objective is determined.

Three major instructional decisions affect the structure of the Word Problem Program. This section presents a rationale for those

decisions, drawing upon instructional research. The three decisions are: (1) the program teaches the translation of word problems into number sentences, but not the solution of number sentences or computational skills; (2) the instructional method combines the elements of traditional methods which have shown greatest success according to the educational literature (Gorman, 1967); and (3) the program sequences the objectives by taking into account the information processing demands of the translation to number sentences.

A. The Role of Computation

Section I.A described the role numeric values in a word problem can play in determining how students respond. In extreme cases, the numbers alone cue the correct operation, and the student responds algorithmically rather than by solving the problem. The order in which problems are presented can radically alter the difficulty of individual problems. For example, both Lofcus (1970) and Hydle and Clapp (1927) found problems are easier when they can be solved by the same operations (in the same order) as the preceding problem. Students often respond to a new problem by attempting to apply the solution steps from the previous problem.

To avoid reinforcing undesirable algorithms such as those described above and in Section I.A, the Word Problem Program employs three strategies. First, the problems presented by the program are written so that no word or phrase consistently cues the correct operation. For example, the words "divided by," "times as many as," "each," and "average" are used in both multiplication and division problems. Second, problems are sequenced so that students never see a set of problems that can all be solved with the same operation. This discourages responding with the operation used in the preceding problem.

Finally, the numbers in the Word Problem Program are chosen so that they vary in magnitude and so that one of the numbers is a factor of another, regardless of which arithmetic operation correctly solves the problem. This feature eliminates the cues that the numbers themselves provide and, along with the first two features, forces the student to find more appropriate strategies for choosing a particular operation.

Section I states that a goal of the Word Problem Program is to teach the translation of a word problem to a number sentence, but not to teach the solution of the equations or the computation of the final answer. For this reason, as well as to allow the freedom to choose numbers by the above criteria, the program must do the arithmetic calculation for the student. The Word Problem Program requires that the student specify the operations and operands necessary to solve the problem, but does not assume that they have the ability to compute with the large numbers used in the problems.

B. Instructional Strategies

The Word Problem Program combines the traditional instructional methods that have shown the greatest success. These include the method of formal analysis, and the method of analogy. The choice of instructional strategies to employ in the Word Problem Program was made on the basis of both the goals of the program and the results of research in the area. The program has dual goals: the development of skills to increase competency in solving arithmetic word problems, and the development of more general problem solving skills. Previous research indicates which methods will be effective in attaining these goals.

Much debate and controversy concerns the question of whether to teach specific word problem solving strategies or algorithms (and if so, which ones), or to teach general problem solving skills and reading skills,

or simply to employ the practice-only method. While yielding contradictory results as to which methods are most effective, research has proved helpful in deciding which methods to employ in the Word Problem Program.

The program emphasizes translating the problem to number sentences. An analytic instructional method would encourage the student to proceed systematically with the translation. Several methods, which are all called analytic, have been studied. For example, Morton (1925) defines his analytic method as a three-step process, whereas Washburne and Osborne (1926) use a six-step process. All definitions, however, include three crucial steps: (1) determine what is to be found; (2) determine what relationships and numerical data are given; and (3) decide which mathematical operations are needed to attain the solution.

In their informative review, Suydam and Riedesel (1969) conclude that "informal procedures are superior to following rigid steps . . . if the analysis method is used, it is recommended that only one or two of the steps be tried with any one problem" (p. 50). Rather than rely totally on analysis, a second method, relying on analogous problems, has been incorporated in the Word Problem Program.

The analogy method has a variety of definitions in the literature. All include two crucial parts: (1) a problem similar to the original is presented by the instructor; and (2) all the information necessary for the correct solution is preserved in the analogous problem (Morton, 1925; Gorman, 1967).

In the Word Problem Program, the analysis method and the analogy method are used when the student asks for a hint. The set of hints for each problem models the translation process. The rest of this section describes the hints in general terms, and gives examples from actual problems.

The first hint for each problem identifies the unknown for the student and suggests that he reread the problem. This hint corresponds to the first step of the analytic method. The second hint is a restatement of the problem in simpler syntactic form, omitting superfluous information. The student is directed to compare the restatement to the original. This second hint corresponds to the analogy method. The third hint is a mathematical statement, relating the variables in the problem, and giving an appropriate number sentence. This hint corresponds to the second step of the analytic method by making the relationships between variables explicit. Since we assume students know how to solve number sentences, no hint corresponds to the third step of the analytic method. Some examples (Figures 2 and 3) will clarify the relationship between hints and the original problem.

The Problem	IN APRIL HARRY HELD SOME TOYS. YESTERDAY, HE DIVIDED THE TOYS INTO 45 BOXES. HOW MANY TOYS DID HE HOLD IN APRIL, IF THERE WERE 3 TOYS IN EACH BOX?
First Hint	HINT: RESTATE THE QUESTION. FIND THE NUMBER OF TOYS HARRY STARTED WITH. NOW REREAD THE PROBLEM.
Second Hint	HINT: RESTATE THE PROBLEM. IT IS SIMILAR TO: HARRY'S TOYS DIVIDED INTO 45 BOXES GIVES 3 TOYS IN EACH BOX. ? = TOYS HARRY STARTED WITH. NOW REREAD THE ORIGINAL PROBLEM.
Third Hint	HINT: TRANSLATE THE PROBLEM TO A NUMBER SENTENCE: ? TOYS / 45 BOXES = 3 TOYS IN A BOX, OR, ? / 45 = 3 NOW, SOLVE THE NUMBER SENTENCE.

Figure 2. A Simple Word Problem and Its Hints.

The Problem	10 GROUPS OF WOMEN AND 19 GROUPS OF PUPILS WENT TO THE BALL PARK. HOW MANY MORE PUPILS THAN WOMEN WERE THERE AT THE BALL PARK, IF THERE WERE 9 PEOPLE IN EACH GROUP?
First Hint	HINT: RESTATE THE QUESTION. FIND THE NUMBER OF MORE PUPILS THAN WOMEN AT THE BALL PARK. NOW REREAD THE PROBLEM.
Second Hint	HINT: RESTATE THE PROBLEM. IT IS SIMILAR TO: 10 GROUPS OF 9 WOMEN AND 19 GROUPS OF 9 PUPILS WERE AT THE BALL PARK. ? = MORE PUPILS THAN WOMEN. NOW REREAD THE ORIGINAL PROBLEM.
Third Hint	HINT: TRANSLATE THE PROBLEM TO A NUMBER SENTENCE: 10 X 9 WOMEN + ? WOMEN = 19 X 9 PUPILS, OR, 10 X 9 + ? = 19 X 9 NOW, SOLVE THE NUMBER SENTENCE.

Figure 3. A Complex Word Problem and Its Hints.

The hint structure used in the program is designed to encourage the student to take the following steps in problem analysis: (1) identify the unknown; (2) translate the word problem into a simpler analogous problem; (3) translate the simpler problem into an arithmetic number sentence; and (4) solve the number sentence. This structure combines the best features of the analytic and analogy methods, providing help when it is needed, and modeling desirable strategies for solving word problems.

C. Defining and Sequencing the Instructional Objectives

Instructional objectives for the Word Problem Program contain groups of problems that require the same kinds of information processing skills. Instructional objectives are sequenced so that later objectives

require quantitatively more or qualitatively different processing skills. This approach to defining objectives and sequencing them is unique to the Word Problem Program, although it depends on two research traditions. Older methods for sequencing and describing objectives seem inadequate for the Word Problem Program.

Sequencing has been long regarded as the crucial issue in teaching students to solve word problems. For example, Goals for School Mathematics: The Report of the Cambridge Conference on School Mathematics (Educational Services, Inc., 1963) states that ". . . problem material should be considered at least as important as the text proper. . . . We therefore believe that the composition of problem sequences is one of the largest and one of the most urgent tasks in curricular development" (p. 28).

1. The Structural Variable Approach

Previous attempts to create instructional objectives with word problems of homogenous difficulty have used "structural variables" to classify problems. Structural variables are easily quantified characteristics of problems which, hypothetically, affect the difficulty of solving the problems. For example, the number of different arithmetic operations required to solve the problem is a frequently used structural variable. It is easily quantified. The hypothesis about its relationship to difficulty can be stated as "Other things being equal, a problem requiring more arithmetic operations is more difficult to solve." The hypothesis is plausible; however, it provides no guidance about why the problems are more difficult. The question of what extra information processing is required to solve a problem with more arithmetic operations is left unanswered. Even the most sophisticated studies of structural variables, using multiple regression techniques, leave unspecified the information

BEST COPY AVAILABLE

processing mechanisms that make the harder problems harder. The rest of this section examines some research on structural variables.

Structural variables are defined as intra- and inter-problem variables whose values are hypothesized to affect the difficulty of a particular word problem. Structural variables such as the number of arithmetic steps to solve the problem, number of words in the problem, and the specific operations required to solve the problem have been investigated extensively. Recent studies were conducted by Loftus and others in the context of a computer program for sixth grade students (Loftus, 1970; Suppes, Loftus, & Jerman, 1969; and Loftus & Suppes, 1972). The objective of their work was to obtain, by using certain structural variables, a regression equation that predicts the probability of a correct response for each problem (Loftus, 1970).

The probability of a correct response, the dependent variable in the Loftus studies, is an unusual variable because problems were presented not on paper, but on a computer terminal and because students were not required to compute answers, but only to indicate what arithmetic the computer should do. Since the Word Problem Program is derived from the Loftus work, the method of presentation is described in some detail below.

Students work at a teletype and see problems printed out for them as shown in Figure 4. The researchers were more interested in the students' ability to solve the problem, not computational facility. The students, therefore, command the computer to perform calculations for them. Every number in the problem is printed on a separate line below the text, and labeled with the letter "G," standing for "given number." Each line is numbered, so the student can refer to the given number (28 in the example) by its line number, (1). Arithmetic operations require two line numbers and the name of the operation.

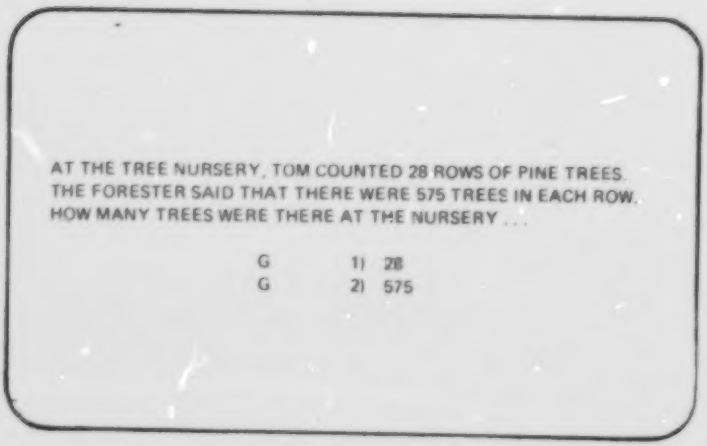
AT THE TREE NURSERY, TOM COUNTED 28 ROWS OF PINE TREES.
THE FORESTER SAID THAT THERE WERE 575 TREES IN EACH ROW.
HOW MANY TREES WERE THERE AT THE NURSERY . . .
G 1) 28
G 2) 575

Figure 4. Loftus Word Problem Format.

In Figure 5, the student has requested that the computer multiply the numbers given on lines 1 and 2 to get a new number. This is indicated by the request "1, 2M." The computer does the multiplication, labels the new line with the next consecutive line number, "3," and is ready to accept a new request. The symbols A, S, M, and Q refer, respectively, to the operations addition, subtraction, multiplication, and division. The correct answer is indicated by the line number and the letter X. The unfortunate choice of notation was apparently dictated by programming considerations, and has been changed in subsequent versions of the program (Suppes, Fletcher, Zanotti, Lorton, & Searle, 1973; Searle, Lorton, & Suppes, 1973).

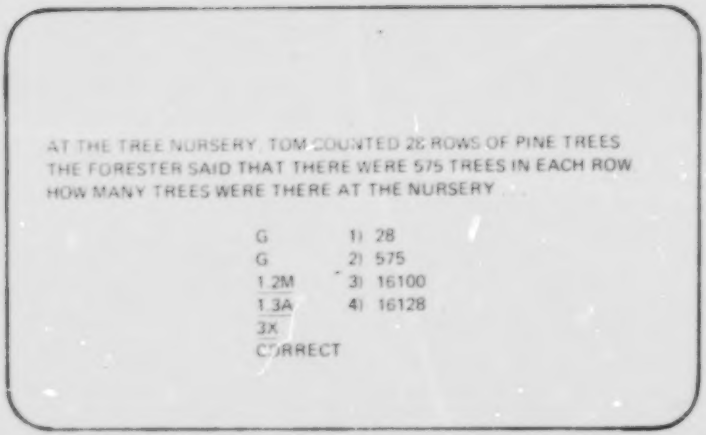

Figure 5. Loftus Response Format

The goal of the Loftus study was to obtain "probability of a correct response" measures to use as the dependent variable on a stepwise regression analysis. One hundred problems were studied. The eight independent variables entered as predictors in the stepwise regression are defined below; significance in the regression is noted for each variable. The variables entered the equation in the order listed, and that influences both the significance and the change in the multiple regression from step to step.

a. The Operations variable was defined as the minimum number of different operations required to solve the problem. A problem solved by adding seven numbers requires one operation whereas a problem solved by averaging two numbers requires two operations. Problems with more operations proved significantly more difficult; the multiple correlation coefficient for the regression for this variable alone was R = .67.

b. The Sequential variable measured a relationship between problems. The value of the sequential variable is either zero or one, depending on whether the problem measured can be solved by the same operations in the same order, as the preceding problem. Problems were significantly easier if they could be solved in this way. The value of R increased to .77.

c. The Length variable was the number of words in the problem and was significant in the regression. The value of R increased to .79.

d. The Depth variable, as defined by Loftus, was a measure of the syntactic complexity of the sentences within a word problem. A number was assigned to each word, depending on how embedded the word was in the sentences' structure diagram, a concept defined by Yngve (1964). The mean of the Yngve numbers for each word in the sentence was computed for every sentence. The largest mean Yngve number, across all the sentences within the problem, became the value of the depth variable. Problems with high values of the depth variable were significantly more difficult. The value of R increased to .81.

e. The Conversion variable was assigned the value one if the student had to convert from one unit of measure to another. The value was zero otherwise. Problems requiring conversion were significantly more difficult. The value of R increased to .83.

f. The Verbal Clue variable was assigned the value one if certain key words occurred in the problem. The key words were hypothesized to make the problem easier to solve. In addition problems, the word "and" was the clue word; in subtraction, "left" was the clue word; in multiplication problems, "each" was the clue word; in division problems, either the word "each" or the word "average" was taken as a verbal clue. This variable did not add significantly to the accuracy of the equation.

g. The Order of Numerical Data variable was assigned the value one in problems where the numbers could be used to solve the problem in the order they appeared. This variable did not add significantly to the accuracy of the equation.

h. The Steps variable was the minimum number of applications of any operation required to reach the solution. Thus, a problem solved by adding seven numbers required six steps, whereas one solved by averaging two numbers required two steps. This variable did not add significantly to the accuracy of the equation.

Several other researchers have investigated structural variables in more traditional settings. Hydle and Clapp (1927) undertook a study involving 7029 subjects. They found that problems with an abstract objective setting (containing situations which are difficult to visualize) and those which contain unfamiliar objects, large numbers, symbols, or superfluous information are significantly more difficult than related problems which do not contain these. (The level of significance was not reported.) The Position of the Question variable was investigated by Williams and McCreight (1965). They found that problems in which the question appeared at the beginning of the problem were significantly easier ($p < .1$) than those in which the question appeared at the end.

Rosenthal (1971) investigated the sequence of information in addition and subtraction word problems based on the following number sentences: $m \pm n = ?$ and $? \pm n = p$. Each problem involved a person "starting out" with "m" objects. Either "m" or "p" was chosen for the unknown quantity. The following results were obtained:

(1) Significant differences were found in performance on problems generated from the four forms ($p < .001$).

BEST COPY AVAILABLE

(2) Problems in which information was presented in normal chronological order were significantly easier than those with the order reversed ($p < .0001$).

For example, for the "$m + n = ?$" form, the problem,

IF PAUL STARTED OUT WITH 5 BOATS AND HE BOUGHT 3 BOATS, HOW MANY BOATS DID HE END UP WITH?

in which the events are mentioned in their proper chronology, was less difficult than

HOW MANY BOATS DID PAUL END UP WITH IF HE BOUGHT 3 BOATS, AND HE STARTED OUT WITH 5 BOATS?

which is stated in reverse chronology.

(3) No significant differences were found between problems due to differential effects associated with the specific problem verb.

Rosenthal's work suggested one of the dimensions for distinguishing among word problems used, namely, the difficulty of problems requiring external rules. (See Section II.C.3 for further discussion.)

The structural variable approach suggests relationships between features of problems and information processing load on the solver; it does not explicate those relationships. The Word Problem Program requires instructional objectives based upon what the student must do. Even the most quantitative work with structural variables, the Loftus studies, does not provide a basis for sequencing a new set of problems. The Loftus study used a fixed set of 100 problems on a population of slow learners; there is reason for skepticism about using the derived equation to rank a new set of problems for a new population of students. Further doubt is cast because the awkward response mode (see Figure 5) took four to eight weeks to master, and may have influenced the dependent variable, probability of a correct response. The next section describes another approach to the study of word problems.

2. The Information Processing Approach

An alternate tradition for investigating word problems has developed in the information processing and artificial intelligence literature. The work has not been concerned with teaching people to solve word problems, but with modeling how they do solve problems. Specifically, the Bobrow (1968) program, STUDENT, is an artificially intelligent computer program that solves algebraic word problems. While Bobrow himself does not claim that humans solve problems in a manner similar to STUDENT, subsequent investigation by Simon and Paige (1966) demonstrated that some humans do indeed perform similarly to STUDENT. Together, these studies make it plausible that the information processing steps required to solve a word problem can be identified, and that instructional objectives can be formed on the basis of those steps.

The STUDENT program processes word problems phrase by phrase, building equivalent algebraic expression piece by piece. The following example traces the translation process, preserving the spirit of the STUDENT program, but using more familiar notation.

> IF THE NUMBER OF CUSTOMERS TOM GETS IS 2 TIMES THE NUMBER OF ADVERTISEMENTS HE RUNS, AND THE NUMBER OF ADVERTISEMENTS IS 45, WHAT IS THE NUMBER OF CUSTOMERS TOM GETS?

First, the problem is partitioned into phrases, and variables are identified, indicated below by parentheses.

> (THE NUMBER OF CUSTOMERS TOM GETS) IS (2) TIMES (THE NUMBER OF ADVERTISEMENTS HE RUNS).
> (THE NUMBER OF ADVERTISEMENTS) IS (45).
> (WHAT) IS (THE NUMBER OF CUSTOMERS TOM GETS)?
>
> Let X = (THE NUMBER OF CUSTOMERS TOM GETS)
>
> Let Y = (THE NUMBER OF ADVERTISEMENTS HE RUNS)

Let Z = (THE NUMBER OF ADVERTISEMENTS)

Now the program builds number sentences step by step.

X = 2Y

Z = 4X

Y = X

These three equations are passed to the solution stage, which recognizes that there are not enough relations to determine the unknown. Solution stage reinvokes translation stage, which tries to make more equations. The first mechanism to make more equations tries to find two distinct variables which name the same entity. It proceeds by matching words within phrases. The similarity of

(THE NUMBER OF ADVERTISEMENTS) to
(THE NUMBER OF ADVERTISEMENTS HE RUNS)

is apparent on a word by word match. Translation stage sets up a new equation (Y = Z), and passes the expanded set to solution stage, which then solves the problem.

Other mechanisms in the program allow STUDENT to recall certain facts it knows, such as "rate times time equals distance," or "three feet equal one yard," or "plural of child equals children." These facts are recalled whenever new equations are required, and appropriate key words appear in the problem. Other mechanisms allow STUDENT to recognize special verbs which indicate time sequences.

The STUDENT program vividly demonstrates the feasibility of algorithmically translating word problems into number sentences. The mechanisms developed to identify phrases, to introduce knowledge external to the problem, and to determine when two phrases refer to the same entity are analogs of the information processing skills humans use to solve similar problems.

While the spirit of the algorithm is directly applicable, the formulation of the algorithm is not. Several factors forced us to develop our own algorithm: (1) STUDENT is an artificial intelligence program and solves problems without regard for human processing capacities. At least for elementary school students, the steps STUDENT takes are too large. (2) STUDENT has a small vocabulary. It could be expanded to recognize and categorize more words, but the effort would be tedious. (3) STUDENT does not output lists of information processing skills which would be useful for categorizing problems. (4) The STUDENT algorithm does not take into account or explain some of the research on structural variables. (5) Problems frequently must be reworded for STUDENT to process them correctly. The next section illustrates an approach which combines structural variables with information processing.

3. A Combined Approach

This section describes the method used in the Word Problem Program for creating instructional objectives, and sequencing the objectives for individual students, a methodology developed by Laudato (in preparation). The method combines features of the structural variable approach with features of the information processing approach. Because ideas in this section are developed in considerable detail, a brief overview follows.

First, the section presents the concept of an "item form," a condensed representation of a set of syntactically similar problems. These item forms are used in the Word Problem Program to reduce storage requirements, and to increase the variety of problems which can be displayed. Next, an example of the information processing algorithm is presented informally. The algorithm analyzes an item form, and outputs data about the processing steps required to translate the problem. Next,

there is a discussion of how the data about processing steps were classified to create instructional objectives based on similarity of information processing demands. The classification results in a three-dimensional array with 24 cells, each corresponding to an instructional objective. The section concludes with a discussion of the use of the array to sequence objectives within the Word Problem Program.

The first task, leading to the creation of instructional objectives, was a definition of the universe of problems. This was accomplished using item forms to specify sets of problems which have identical syntactic form (Osburn, 1968; Hively, 1963). Various grammatical slots in the item form are filled by any word on a named list. Figure 6 contains an example of an item form and its associated lists of words, as well as two different problems generated from the item form. Each phrase in square brackets ([ ]) stands for a randomly selected word from the appropriate list. The item form in Figure 6 could appear with any of 120 different combinations of MALE NAMES, OBJECT WORDS, OWNED WORDS, and FEMALE NAMES (2 X 5 X 4 X 3 = 120). In the Word Problem Program, lists contain five to 20 elements and individual problems utilize five to ten different lists, considerably increasing the number of individual problems generated from an item form. The randomly selected numbers further diversify the appearance of the problem. However, in spite of the apparent diversity, each of the many problems has the same syntactic structure, and can be translated into a number sentence by the same processes.

Item Form	[MALE NAME] [OWNED WORD] [NUMBER 1] [OBJECT WORD]. IF HE [OWNED WORD] [NUMBER 2] TIMES AS MANY [OBJECT WORD] AS [FEMALE NAME], HOW MANY [OBJECT WORD] DID THEY HAVE TOGETHER?
Lists	OBJECT WORDS = {bananas, oranges, trucks, stamps or farms } MALE NAMES = {Dick or Nick } FEMALE NAMES = {Susan, Joan or Mary Ellen } OWNED WORDS = {had, owned, bought or found }
Example 1	Nick had 75 stamps. If he had 5 times as many stamps as Joan, how many stamps did they have together?
Example 2	Dick bought 1500 oranges. If he bought 60 times as many oranges as Mary Ellen, how many oranges did they have together?

Figure 6. Example of an Item Form.

A collection of 200 item forms defines the curriculum covered by the Word Problem Program. The problems range in difficulty from approximately second grade to junior high school. The range could be expanded, but is sufficiently extensive for use in an elementary school.

An algorithm was designed to process the item forms and output lists of data about the information processing steps required to translate the item forms to number sentences. The algorithm is not sufficiently complete to be run on a computer, but has been hand simulated. The next paragraphs give the flavor of the algorithm as it would apply to the item form in Figure 6. To make the notation more concrete, the wording from example 1 is used rather than the list names. The steps in the translation process are listed and numbered in Figure 7.

Example 1 with phrases marked		(Nick had 75 stamps.) If (he had 5 times as many stamps as Joan), (how many stamps did they have together?)
Initial translation of phrases	(1) (2) (3)	(stamps, Nick) = 75 (stamps, he) = 5· (stamps, Joan) ? = (stamps, they)
Identification of pronouns	(4) (5)	he = Nick they = Nick and Joan
Insert identities into initial phrases	(6) (7)	(stamps, Nick) = 5· (stamps, Joan) ? = (stamps, Nick and Joan)
Apply union rule (8) to (7)	(8) (9)	(A,X and Y) = (A,X) + (A,Y) ? = (stamps, Nick + (stamps, Joan)
Apply indirect rule (10) to (6)	(10) (11)	(X,Y) = A· (X,Z) if and only if (X,Y) / A = (X,Z) (stamps, Nick) / 5 = (stamps, Joan)
Substitute (11)	(12)	? = (stamps, Nick) + (stamps, Nick) / 5
Substitute (1)	(13)	? = 75 + (stamps, Nick) / 5
Substitute (1)	(14)	? = 75 + 75 / 5

Figure 7. Steps in Translating an Item Form.

The first step in the algorithm is to mark off the individual phrases for translation. Phrases end with commas, periods, or sometimes the word "and." Example 1 has three phrases. Next, each phrase which assigns a value to a quantity is translated to assignment notation. For example, "Nick had 75 stamps" is translated as (stamps, Nick) = 75. In general, the notation (A, B) = N can be interpreted as "set the number of A of B to N." All three phrases in the example are translated to assignments.

In the next phase, the algorithm identifies the pronouns with their referents. Since the item form specifies male and female names, this

process is straightforward. Identifications are made in lines (4) and (5) of Figure 7, and inserted into phrases (1) and (2) in lines (6) and (7).

The word "and" in an assignment (line 7) cues the use of the "union rule" on line (8). The union rule transforms an assignment to a union to the sum of two simpler assignments. The use of this external rule (or any of several others) is noted on the list of information processing required for translation.

Next, a check is made for available numeric substitutions for all variables in line (9). Line (1) provides a substitution for (stamps, Nick), but no substitution is available for (stamps, Joan). The only reference to (stamps, Joan) is in line (6), and it must be transformed by the indirect rule (line 10). The indirect rule is analogous to using the multiplicative inverse. The use of one of the indirect rules is noted on the list of information processing required for translation.

Finally, each phrase is available in direct form, and three simple substitutions are performed on line (9) to create line (14), the number sentence for the problem. The fact that three substitutions were required is noted in the list of information processing required for translation.

The algorithm operates in similar fashion on each item form used in the Word Problem Program to produce lists of information processing steps required for translation. In addition to the notations generated in the example, the algorithm also lists the use of rules stated explicitly in the problem, the use of unstated formulae, the use of conversion information, and the generation of assignments that are superfluous to the eventual translation. The output lists for each of the 200 item forms were created and examined for similarities and differences. Two distinctions emerged easily: First, half the problems required one or more indirect

rules in the translation, and half did not; the second distinction was based on the number of substitutions required to translate the item form. More substitution implies more processing and greater difficulty. Item forms were classified into three sets--those with one or two substitutions, those with three substitutions, and those with four or five substitutions. The values were chosen to keep nearly the same number of item forms in each of the three sets. This second distinction into three sets crossed the direct-indirect distinction to give six sets of item forms. Problems within a set were still far from homogenous with respect to information processing load, so a further distinction was required.

After several preliminary attempts, four categories emerged: (1) item forms which required only substitutions, (2) item forms which required special transformation rules, (3) item forms which required external knowledge to complete the translation, and (4) item forms which contained numeric phrases that proved superfluous to the translation. Since these four categories are not mutually exclusive, a classification priority was established: Superfluous data preempted any other classification; in the remaining item forms, external knowledge preempted other classifications; and the remaining item forms either required special transformation rules or could be translated using substitution only. The four categories crossed with the six existing sets produced 24 sets of item forms.

Within any of the 24 sets, the quantity and kind of information processing skills are homogenous, precisely the criterion for instructional objectives. The three classification attributes can be represented in the three-dimensional array of Figure 8. Every item form fits into one of the instructional objectives represented in the array. For example, the analysis in Figure 7 of one item form indicates that an indirect

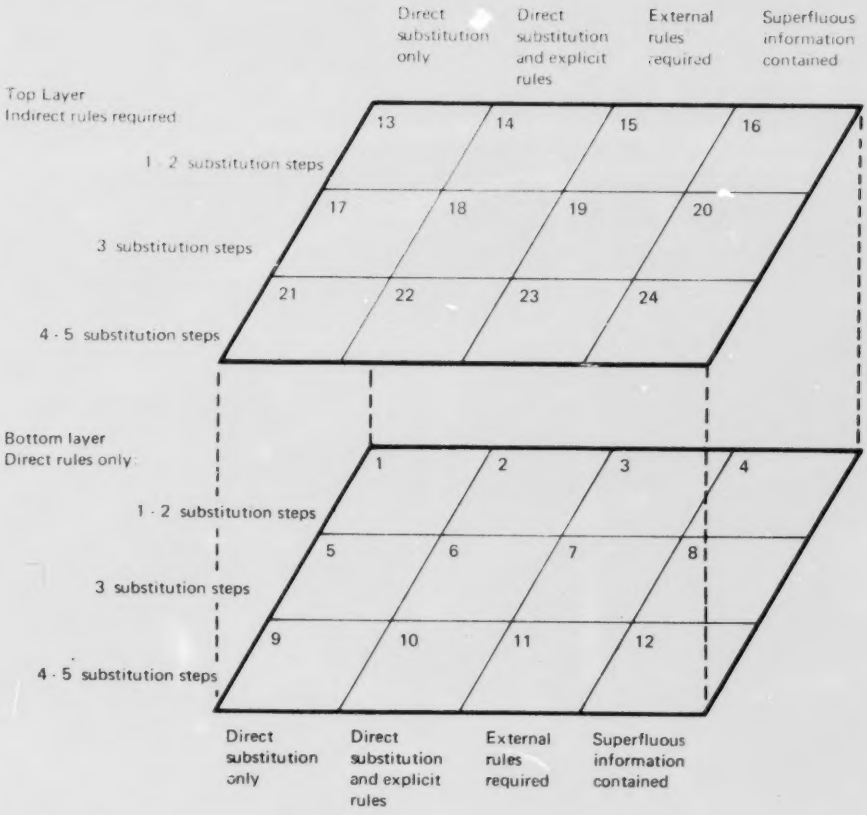

Figure 8. Array of Instructional Objectives for Problem Types.

transformation was used (line 10), placing the item form in the top layer (objectives 13-24). Three substitutions were required (lines 12-14), placing the item form in the second row (objectives 17-28). No superfluous information was included, but the "union rule" was required (line 8), placing the item form in the third column, (objective 19).

Since the instructional objectives in the array differ in only small ways from their neighbors, the array is used to make sequencing decisions. The decisions are based on these hypothesized relationships among cells in the array: (1) An objective in the top layer, requiring an indirect rule for translation, is more difficult than the corresponding objective without the indirect rule. For example, objective 1 is easier than objective 13. (2) Within a layer, objectives in the same column which require fewer substitution steps are easier. For example, objective 1 is easier than objective 5 which is easier than objective 9. (3) Within a layer, objectives in the same row have a more complex relationship since no known relationship exists between the use of superfluous information and the use of external rules. As an example, objective 1 is easier than objective 2, which is easier than both objectives 3 and 4. No special relationship is hypothesized between objectives 3 and 4.

Sequencing of instructional objectives in the Word Problem Program is based on these hypothesized difficulty relationships. Students progress from row to row within a column, across a column, or from the bottom layer to the top layer. Whenever one objective is mastered, the next is selected to be more difficult than some previously mastered objective along one of the dimensions of the array. The choice of objective is further constrained in that no objective can be considered available unless all of the immediately easier objectives have been previously mastered. These constraints still allow students to progress along different paths, depending on their learning histories. A student who has

difficulty progressing from bottom to top layer objectives will normally move along rows or columns first.

An example of the sequencing decisions for one student's mastery pattern is represented in Figure 9. Objectives labeled "mastered" are within the student's repertoire. Objective 4, labeled "attempted," indicates the student failed to master that cell in a recent session. Objectives labeled "available" are those for which hypothesized prerequisites have been mastered, but which have not been recently attempted. The next instructional objective selected will be one of the three objectives marked available.

Section II discussed three instructional design decisions implemented in a preliminary version of the Word Problem Program and pilot tested in Spring 1974. Section III describes that program and how it embodies the three design decisions of this section that include: allowing the student to indicate the solution without doing calculations; making use of the analytic and analogy instructional methods; and presenting instructional objectives based on the information processing demands of the problems.

Section III: The Word Problem Program

This section describes how a student interacts with the Word Problem Program. The program was implemented to embody the design principles described in Section II. Section III.A is based on the pilot version of the program, operational in Spring 1974. Section III.B describes the complete version planned for Fall 1974.

BEST COPY AVAILABLE

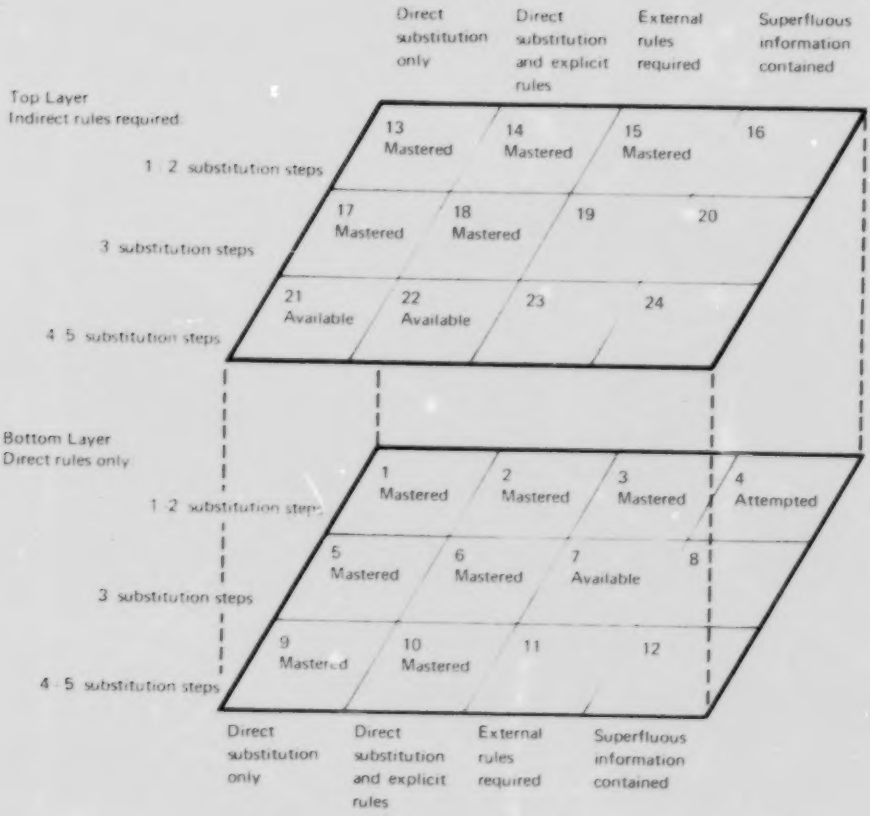

Figure 9. One Student's Mastery Pattern

A. The Pilot Version, Spring 1974

A student's interaction with the Word Problem Program begins with a problem written on the screen of a cathode-ray tube, and the numbers contained in the problem displayed on labeled lines below the problem. An initial display appears in Figure 10.

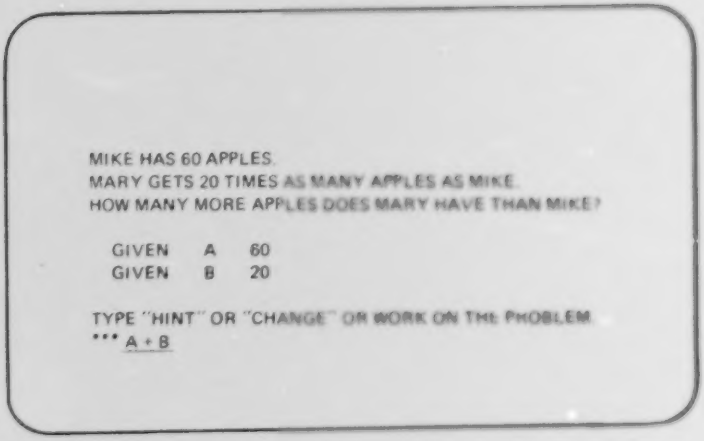
MIKE HAS 60 APPLES.
MARY GETS 20 TIMES AS MANY APPLES AS MIKE.
HOW MANY MORE APPLES DOES MARY HAVE THAN MIKE?
GIVEN A 60
GIVEN B 20
TYPE "HINT" OR "CHANGE" OR WORK ON THE PROBLEM.
*** A + B

Figure 10. An Initial Problem Display.

After reading the problem, the student has several options: He can type "CHANGE," and receive a new problem; he can type "HINT," and be given a hint; he can type any two line names (A or B in Figure 10) separated by an arithmetic operation symbol (+, -, X, or /), and have the computer do a calculation; he can type "ANSWER" and a line name, and be told if that line contains the correct answer; or, finally, he can type "DEFINE" and a number, and have the computer create a new line name with that numerical value. In Figure 10, the student typed "A + B."

indicating that he wants to create a new line, whose value is the sum of the number on line A (60) and the number on line B (20). The computer does the calculation, erases the instructions from the display, and creates a new line. The resulting display is in Figure 11.

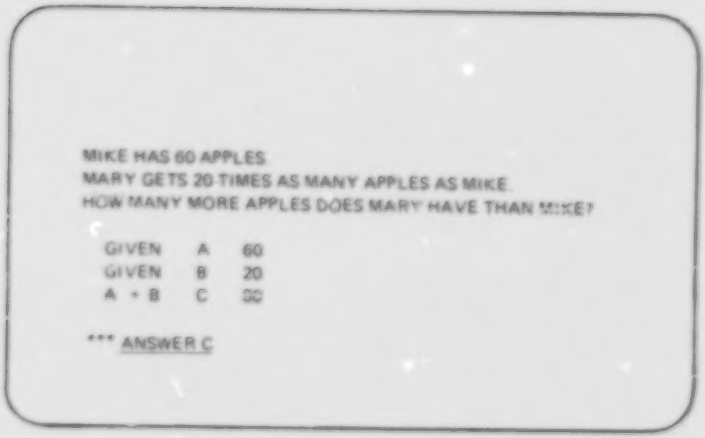
MIKE HAS 60 APPLES.
MARY GETS 20 TIMES AS MANY APPLES AS MIKE.
HOW MANY MORE APPLES DOES MARY HAVE THAN MIKE?
GIVEN A 60
GIVEN B 20
A + B C 00
*** ANSWER C

Figure 11. Display After One Computation.

The student in Figure 11 claims that the correct answer is on line C. The computer informs him that the answer is incorrect, and suggests that he ask for a hint. The resulting display appears in Figure 12.

```
MIKE HAS 60 APPLES
MARY GETS 20 TIMES AS MANY APPLES AS MIKE.
HOW MANY MORE APPLES DOES MARY HAVE THAN MIKE?

  GIVEN    A    60
  GIVEN    B    20
  A * B    C    80
ANSWER     C    INCORRECT ANSWER

TYPE "HINT" IF YOU WANT SOME HELP
*** HINT
```

Figure 12. An Incorrect Answer Display.

In Figure 12, the student took the suggestion and asked for a hint. The computer displays the first hint which identifies the unknown and suggests rereading the problem. The hint display appears in Figure 13.

```
MIKE HAS 60 APPLES
MARY GETS 20 TIMES AS MANY APPLES AS MIKE.
HOW MANY MORE APPLES DOES MARY HAVE THAN MIKE?

  GIVEN    A    60
  GIVEN    B    20
  A * B    C    80
ANSWER     C    INCORRECT ANSWER

HINT: RESTATE THE QUESTION. FIND THE NUMBER OF
MORE APPLES MARY HAS THAN MIKE.
NOW REREAD THE PROBLEM.
*** A X B
```

Figure 13. A Hint Display.

Section II. B described the hint structure, and the reader can refer to Figures 2 and 3 for details.

In Figure 13, the student continues work on the problem by multiplying line A by line B. After further calculation steps, which are not shown, the student indicates the correct answer. The resulting display is contained in Figure 14.

```
MIKE HAS 60 APPLES.
MARY GETS 20 TIMES AS MANY APPLES AS MIKE.
HOW MANY MORE APPLES DOES MARY HAVE THAN MIKE?

  GIVEN    A    60
  GIVEN    B    20
  A * B    C    80
ANSWER     C    WRONG ANSWER

HINT: RESTATE THE QUESTION: FIND THE NUMBER OF
MORE APPLES MARY HAS THAN MIKE.

  A X B    D    1200
  D - A    E    1140
ANSWER     E    GREAT!

1140 IS THE NUMBER OF MORE APPLES
MARY HAS THAN MIKE.
```

Figure 14. A Correct Answer Display.

The interaction format described so far demonstrates how the first two design decisions were implemented. The student does not calculate himself, but indicates the calculation he wishes done. He is expected to indicate the calculations in an order which solves the appropriate number sentence. Analytic and analogy methods are used within the hint structure. Since hints are available, but not compulsory, the student decides when to use them.

The third instructional decision, sequencing problems by quantitative and qualitative differences in information processing load, is not visible in a short interaction. Internally, the program contains descriptions of the information processing characteristics of each problem and selects problems for appropriate objectives. In the pilot version, intervention from the experimenter is required whenever the student has mastered an objective and should begin work on a more advanced objective. The procedures for those decisions have not been automated.

B. The Complete Version, Fall 1974

Pilot testing during Spring 1974 demonstrated that students master the interaction format within 15 minutes, that students enjoy work on the program, and that students become more competent as they work. A detailed account of the pilot testing of the Word Problem Program will be published later.

The expanded version of the Work Problem Program for school use will differ in three important respects from the pilot version: (1) The computer will accept algebraic expressions as directed by the student. This is an expansion of current capabilities and separates the translation stage more completely from the solution stage. (2) The computer will allow students to type numbers rather than line names, if they prefer that form. (3) The computer will make all instructional decisions without experimenter intervention. These three changes are intended to make the program more consistent with the goals, and to provide for individualized instruction without teacher intervention.

This report describes the design and rationale for a computer program which teaches word problem solving skills and promotes general problem solving skills. It distinguishes the translation stage from the

solution stage and focuses attention on teaching the translation stage. Educational and psychological literature are reviewed and used to support three instructional decisions. Finally, a student interaction with the Word Problem Program is described with attention given to how the program embodies the three instructional decisions.

The Word Problem Program will be the basis of further work in the area of word problems. We intend to formalize the algorithm for examining item forms and listing information processing steps. This work may in turn lead to changes in the array of instructional objectives. Finally, analysis of students' work on the program and on independent measures of word problem skills will test the hypothesis that they learn to solve word problems by working on the Word Problem Program. Additional tests will measure changes in general problem solving skills.

References

Bobrow, D. G. Natural language input for a computer problem-solving system. In M. Minsky (Ed.), Semantic information processing. Cambridge: Massachusetts Institute of Technology, 1968.

Educational Services, Inc. Goals for school mathematics. The report of the Cambridge Conference on School Mathematics. Boston: Houghton Mifflin, 1963.

Fitzhugh, R. J. LRDC experimental time-sharing system reference manual (3 vols.). Pittsburgh: University of Pittsburgh, Learning Research and Development Center, 1970.

Gorman, C. J. A critical analysis of research on written problems in elementary school mathematics. Unpublished doctoral dissertation, University of Pittsburgh, 1967.

Hively, W., Jr. Defining criterion behavior for programmed instruction in elementary mathematics. Unpublished doctoral dissertation, Harvard University, 1963.

Hydle, L. L., & Clapp, F. L. Elements of difficulty in the interpretation of concrete problems in arithmetic. Madison: University of Wisconsin, Bureau of Educational Research, 1927. (Research Bulletin, No. 9.)

Laudato, N. C. The design and assessment of a computer-assisted instructional program in word problem solving for the elementary school. Doctoral dissertation, University of Pittsburgh, in preparation.

Lindvall, C. M., & Bolvin, J. O. Individually prescribed instruction: The Oakleaf Project. Pittsburgh: University of Pittsburgh, Learning Research and Development Center, 1966. (Working Paper 8.)

Loftus, E. J. F. An analysis of the structural variables that determine problem solving difficulty on a computer-based teletype. Stanford: Stanford University, Institute for Mathematical Studies in the Social Sciences, 1970. (Technical Report No. 162.)

Loftus, E. J. F., & Suppes, P. Structural variables that determine problem solving difficulty in computer-assisted instruction. Journal of Educational Psychology, 1972, 63(6), 531-542.

Morton, R. L. An analysis of errors in the solution of arithmetic problems. Educational Research Bulletin. Columbus: Ohio State University, April 1925.

Newell, A., & Simon, H. A. Human problem solving. Englewood Cliffs, N. J.: Prentice-Hall, 1972.

Olton, R. M. A self-instructional program for developing productive thinking skills in fifth and sixth-grade children. Journal of Creative Behavior, 1969, 3, 16-25.

Osburn, H. G. Item sampling for achievement testing. Educational and Psychological Measurement, 1968, 28, 95-104.

Papert, S. Teaching children thinking. In Mathematics teaching. Leicester, England: The Association of Teachers of Mathetics, 1972.

Polya, G. How to solve it. Garden City, N. Y.: Doubleday, 1957.

Polya, G. Mathematical discovery. New York: Wiley, 1962.

Riedesel, C. A. Guiding discovery in elementary school mathematics. New York: Appleton-Century-Crofts, 1967.

Rosenthal, D. J. A. The sequence of information in arithmetic word problems. Master's thesis, University of Pittsburgh, 1971.

Searle, B. W., Lorton, P., & Suppes, P. Structural variables affecting CAI performance on arithmetic word problems of disadvantaged and deaf students. Stanford: Stanford University, Institute for Mathematical Studies in the Social Sciences, 1973. (Technical Report No. 213.)

Simon, H. A., & Paige, J. M. Cognitive processes in solving algebra word problems. In B. Kleinmintz (Ed.), Problem solving. New York: Wiley, 1966.

Suppes, P., Fletcher, J. D., Zanotti, M., Lorton, P., & Searle, B. W. Evaluation of computer-assisted instruction in elementary mathematics for hearing-impaired students. Stanford: Stanford University, Institute for Mathematical Studies in the Social Sciences, 1973. (Report No. 200.)

Suppes, P., Loftus, E. J. F., & Jerman, M. Problem-solving on a computer based teletype. Stanford: Stanford University, Institute for Mathematical Studies in the Social Sciences, 1969. (Technical Report No. 141.)

Suydam, M., & Riedesel, C. A. Interpretative study of research and development in elementary school mathematics (3 vols.). Final Report. College Park: Pennsylvania State University, 1969.

Washburne, C., & Osborne, R. Solving arithmetic problems. II. The Elementary School Journal, 1926, 27, 296-304.

Williams, M. H., & McCreight, R. W. Shall we move the question? The Arithmetic Teacher, 1965, 12, 418-421.

Wooten, W. SMSG: The making of a curriculum. New Haven: Yale University Press, 1965.

Yngve, V. H. Implications of mechanical translation research. Proceedings of the American Philosophical Society, 1964.